



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/692,280	10/23/2003	Bryan Thomas Starbuck	MSFT121734	7146
26389 7590 08/20/2007 CHRISTENSEN, O'CONNOR, JOHNSON, KINDNESS, PLLC 1420 FIFTH AVENUE SUITE 2800 SEATTLE, WA 98101-2347			EXAMINER KUMAR, ANIL N	
			ART UNIT 2174	PAPER NUMBER
			MAIL DATE 08/20/2007	DELIVERY MODE PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

## Office Action Summary

Application No.

10/692,280

Applicant(s)

STARBUCK ET AL.

Examiner

Anil N. Kumar

Art Unit

2174

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 14 Jun 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-24 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-24 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- ☐ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☐ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- ☐ Notice of Informal Patent Application
- ☐ Other: \_\_\_\_\_

### **DETAILED ACTION**

1. This action is in response to the amendment filing of June 14th, 2007. Claims (1-24) are pending and have been considered below.

#### ***Claim Rejections - 35 USC § 112***

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. The amended Claim 1 recites the limitation " the uniquely identified insert location being determined by the host in accordance with one or more other commands" in lines 5-6. There is insufficient antecedent basis for this limitation in the claim.

#### ***Claim Rejections - 35 USC § 102***

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 6, 15-17, and 21 are rejected under 35 U.S.C. 102(b) as being unpatentable over Padawer et al. (US 5,220,675).

Claim 6: Padawer et al. disclose a method for the method comprising:

- loading at least one extension (i.e. ... item 1218 updates the CMI array with values returned from Add\_Atom ... col 7 lines 10-15 and Fig. 12), an extension being an application capable of extending the functionality of the host (see Command and arguments in the CMI entry in Fig. 10A);
- uniquely identifying for the at least one extension an available insert location where the extension may request to insert commands into a host UI (i.e. ...whether the CMI array location or slot is available... col 6 lines 61-65 and Fig. 12); and
- obtaining from the at least one extension at least one uniquely identified command to insert at the available insert location; and integrating the command in the available insert location in accordance with the extension's load order.(i.e. ... see flow diagram for Add\_Atom routine... col 7 lines 19-29 and Fig. 13).

Claim 15: Padawer et al. disclose a system to safely modify a host user interface (UI) with extensions, the system comprising:

- a memory component for storing a command item associated with a command from an extension, an extension being an application capable of extending the functionality of the host, (i.e. ... array 1002 is used to store a an array of command menu items... col 5 lines 6-9 and Fig. 10A) and an insert location available to integrate the command into a host UI (i.e. ....

Determine whether a CMI array location or slot is available... col 6 lines 61-65 and Fig. 12);

- a processing unit operable to return the command item in response to a request for commands to integrate into the host UI at the insert location (i.e. ... a flow diagram to "Add" utility 1128 which is invoked by the routine 1110 to create a new entry in the CMI array 1002... col 6 lines 59-61 and Fig. 12); . and
- a display device to display the command from the extension integrated into the host UI at the insert location in accordance with command item UI (i.e. ...displaying the user customizable interface.. col 2 lines 37-39 and Figs. 1-9).

Claim 16: Padawer et al. disclose system to safely modify a host user interface (UI) with extensions, as in claim 15 above. Furthermore Padawer et al. disclose further wherein the processing unit is operable to integrate multiple commands into the host UI at the insert location in accordance with a priority of an extension that generated the commands (i.e. ... a flow diagram to "Add" utility 1128 which is invoked by the routine 1110 to create a new entry in the CMI array 1002... col 6 lines 59-61 and Fig. 12).

Claim 17: Padawer et al. disclose system to safely modify a host user interface (UI) with extensions, as in claim 15 above. Furthermore Padawer et al. disclose wherein the processing unit is operable to determine the priority of the extension

that generated the command based on an order in which the extension was loaded into the system (i.e.... array means are provided for storing data structures... note by definition, the display of CMI entries in an array will be of the same priority as they were stored... col 2 lines 5-10 and Fig 10A).

Claim 21: Padawer et al. disclose components for safely modifying a host user interface with an extension user interface, the medium comprising:

- a user interface (UI) resource having a command item data structure in which to store a command UI, and an insert location data structure to store an available insert location in a host UI (i.e. ... array 1002 is used to store a an array of command menu items... col 5 lines 6-9 and Fig. 10A);
- and a host interface to expose the available insert locations to extensions that have commands to insert into the host UI (i.e. .... Determine whether a CMI array location or slot is available... col 6 lines 61-65 and Fig. 12), and to receive a count of command items and command items representing the extension's commands, (i.e. ... determines the number of menu items... col 6 35-37 and Fig. 11) an extension being an application capable of extending the functionality of the host UI (i.e. command /arguments in CMI entry, Fig. 10A);
- an extension interface to receive the available insert locations from the host and to provide the count of command items and command items representing the extension's commands (i.e. .... Determine whether a CMI array location or slot is available... col 6 lines 61-65 and Fig. 12); and

- a host process to integrate the extension's command into the host interface in accordance with the UI resource (i.e. ... a flow diagram to "Add" utility 1128 which is invoked by the routine 1110 to create a new entry in the CMI array 1002... col 6 lines 59-61 and Fig. 12).

6. Claims 11-14 are rejected under 35 U.S.C. 102(b) as being unpatentable over Ezekiel at al. (US 5, 625, 783).

Claim 11: Ezekiel at al. disclose a method of communicating between a host and an extension, the method comprising:

- an extension storing a UI for a command in a command item in preparation for communicating with a host (i.e. ... each entry of command table can contain , for example, name of the command,...and a unique identifier... col 6 lines 34-43);
- a host issuing a call to the extension to return a number of command items to be inserted into an insert location (i.e. see the process of Shell –Host- which does similar steps in creating and display the new menu items form the registered command items. Col 9 lines 1-13 and Fig. 6);
- the extension returning the number of command items to be inserted into the insert location (i.e. ... see package registration process... col 8 lines 12-28 and Fig. 4);

- the host issuing a call to an extension to return the command items to be inserted into the insert location (i.e. see the process of Shell –Host- which does similar steps in creating and display the new menu items form the registered command items. Col 9 lines 1-13 and Fig. 6);
- the extension returning the command items to be inserted into the insert location (i.e. ...see package registration process... col 8 lines 12-28 and Fig. 4); and
- the host integrating the command for each of the returned command items into a host UI in accordance with each command item's stored UI. (i.e. see the process of Shell –Host- which does similar steps in creating and display the new menu items form the registered command items. Col 9 lines 1-13 and Fig. 6).

Claim 12: Ezekiel at al. disclose a method of communicating between a host and an extension, as in claim 11 above. Furthermore Ezekiel at al. disclose further comprising: the host issuing a call to the extension to return a status of the command when the integrated command has become visible in the host UI; and the extension issuing a call to the host returning the status of the command; and the host modifying a display of the integrated command in accordance with the returned status of the command (i.e. ... the organizational plan provides a guideline as to which menus are to be included in the menu bar and in what order... col 9 lines 14-24 and Fig. 7).



Claim 13: Ezekiel at al. disclose a method of communicating between a host and an extension, as in claim 12 above. Furthermore Ezekiel at al. disclose further wherein the integrated command has become visible in the host UI when the integrated command is displayed in a menu item in the host UI, and the menu item is being dropped down in response to a user action (i.e. ... the organizational plan provides a guideline as to which menus are to be included in the menu bar and in what order... col 9 lines 14-24 and Fig. 7).

Claim 14: Ezekiel at al. disclose a method of communicating between a host and an extension, as in claim 11 above. Furthermore Ezekiel at al. disclose further comprising: the host issuing a call to the extension to execute the command when the integrated command has been invoked in the host UI; and the extension issuing a call to the host returning the result of the executed command; and the host modifying a display of the integrated command in accordance with the returned result of the executed command (i.e.... shell 300 can send a message to the package object... col 8 lines 60-65).

### ***Claim Rejections - 35 USC § 103***

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 1- 5, 7- 8, 18-20, and 22-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Padawer et al. (US 5,220,675) in view of Ezekiel at al. (US 5, 625, 783).

Claim 1: Padawer et al. disclose a method for modifying (customizing a host user interface (UI)(i.e. customize a menu), the method comprising:

- storing a UI for a command in a uniquely identified command item (i.e. ... array 1002 is used to store a an array of command menu items... col 5 lines 6-9 and Fig. 10A);
- obtaining from a host a uniquely identified insert location available to insert the command's stored UI into the host's UI (i.e. .... Determine whether a CMI array location or slot is available... col 6 lines 61-65 and Fig. 12), the uniquely identified insert location being determined by the host in accordance with one or more other commands; and
- requesting the host to insert the command's stored UI into the host UI at the uniquely identified insert location (i.e. ... a flow diagram to "Add" utility 1128 which is invoked by the routine 1110 to create a new entry in the CMI array 1002... col 6 lines 59-61 and Fig. 12).

but does not disclose a method for uniquely identification of command items.

However, Ezekiel at al. disclose a method for adding a unique identifier to

command items (i.e. ...a unique identifier, for example a integer that uniquely specifies the command... col 6 lines 33-43).

Therefore, it would have been obvious to one having ordinary skill in the art at the time of this invention to add a unique identifier for CMI array entries, in Padawer et al. One would be motivated to uniquely identify all the command items (menu items) especially when the application/extension has to work between different menus of host application.

Claim 2: Padawer et al. and Ezekiel at al. disclose a method for modifying a host user interface (UI), as in claim 1 above. Furthermore, Ezekiel at al. disclose storing a UI for the command in a uniquely identified command item further comprises:

- generating a unique universal identifier (UUID) for the command (i.e. ... identifier may be automatically generated by the computer... col 6 lines 53-58); and
- storing at least one of a text, an icon, a status text, and a hotkey together with the UUID in a command item data structure (i.e. ... each entry of command table can contain , for example, name of the command,...and a unique identifier... col 6 lines 34-43).

Therefore, it would have been obvious to one having ordinary skill in the art at the time of this invention to generate a unique identifier for each command items and store it in CMI array entries, in Padawer et al. One would be motivated to

uniquely identify all the command items (menu items) especially when the application/extension has to work between different menus of host application.

Claim 3: Padawer et al. and Ezekiel at al. disclose a method for modifying a host user interface (UI), as in claim 2 above. Furthermore, Ezekiel at al. disclose, wherein storing a UI for the command in a uniquely identified command item further comprises: obtaining a related command from a sub-extension; storing an index to the related command in the command item data structure; and requesting the host to insert the related command's stored UI into the host UI at the uniquely identified insert location, wherein the related command's stored UI is in its own command item data structure accessible to the host via the index (i.e. ... each module/extension can have one or more associated subsystems / sub-extensions, each of which have associated commands.... Col 5 lines 50-56 and Fig. 3A). Therefore, it would have been obvious to one having ordinary skill in the art at the time to have extension/subsystems associated with commands, in Padawer et al. One would be motivated to associate extension/subsystems with commands especially when the application/extension has to work between different menus of host application.

Claim 4: Padawer et al. and Ezekiel at al. disclose a method for modifying a host user interface (UI), as in claim 3 above. Furthermore, Ezekiel at al. disclose, wherein the related command is one of a sibling of the command and a child of

the command (i.e. ... Module commands or Subsystem commands... Fig. 3A). Therefore, it would have been obvious to one having ordinary skill in the art at the time to have commands associated with extension/subsystems, in Padawer et al. One would be motivated to associate extension/subsystems with commands especially when the application/extension has to work between different menus of host application.

Claim 5: Padawer et al. and Ezekiel at al. disclose a method for modifying a host user interface (UI), as in claim 1 above. Furthermore, Padawer et al. disclose, wherein obtaining from the host the uniquely identified insert location includes requesting from the host available insert locations and receiving from the host at least one UUID identifying each of the available insert locations (i.e. ... array location or slot is available... col 6 lines 61-65 and Fig. 12).

Claim 7: Padawer et al. and Ezekiel at al. disclose a method for modifying a host user interface (UI), as in claim 1 above. Furthermore, Padawer et al. disclose, wherein the uniquely identified command item specifies a UI to be used when integrating the command into the host UI (i.e. ... a flow diagram to "Add" utility 1128 which is invoked by the routine 1110 to create a new entry in the CMI array 1002... col 6 lines 59-61 and Fig. 12), but does not disclose a method for uniquely identification of command items. However, Ezekiel at al. disclose a method for adding a unique identifier to command items (i.e. ... a unique

identifier, for example a integer that uniquely specifies the command... col 6 lines 33-43). Therefore, it would have been obvious to one having ordinary skill in the art at the time of this invention to uniquely identified commands, in Padawer et al. One would be motivated to uniquely identify all the command items (menu items) especially when the application/extension has to work between different menus of host application.

Claim 8: Padawer et al. and Ezekiel at al. disclose a method for modifying a host user interface (UI), as in claim 1 above. Furthermore, Padawer et al. disclose, wherein the specified UI includes at least one of display text, a display icon, a status display text, a help text, an accelerator key, and an index to any related commands, and wherein integrating the command in the available insert location of the host UI is performed in accordance with the specified command UI (i.e. ... see flow diagram Add utility 1128... Col 6 lines 59-62 and Fig. 12).

Claim 18: Padawer et al. disclose a system to safely modify a host user interface (UI), as in claim 17 above. Furthermore, Padawer et al. disclose, wherein the memory component for storing the command item and insert for each of the command item and insert location to uniquely identify the command item and insert location from among a plurality of command items and insert locations (i.e. ... each entry of command table can contain, for example, name of the command, ...and a unique identifier... col 6 lines 34-43), but do not disclose

storing a universally unique identifier (UUID) with the insert location and command item. However, Ezekiel et al. disclose a system that store a universally unique identifier (UUID) with the insert location and command item (i.e. ... each entry of command table can contain , for example, name of the command,... and a unique identifier... col 6 lines 34-43). Therefore, it would have been obvious to one having ordinary skill in the art at the time of this invention to store a unique identifier for each command items and insert locations, in Padawer et al. One would be motivated to uniquely identify all the command items (menu items) and insert locations especially when the application/extension has to work between different menus of host application.

Claim 19: Padawer et al. disclose a system to safely modify a host user interface (UI), as in claim 18 above, but does not disclose wherein the processing unit is operable to request a command status from the extension that generated the command by the command's UUID. However, Ezekiel et al. disclose a system that uses universally unique identifier (UUID) or group identifiers to manipulate menu items (i.e. ... group identifiers are used in sorting commands... col 6 lines 50-53). Therefore, it would have been obvious to one having ordinary skill in the art at the time of this invention to use a unique identifier to manipulate menu items, in Padawer et al. One would be motivated to use unique identifiers to manipulate all menu items especially when the application/extension has to work between different menus of host application.

Claim 20: Padawer et al. disclose a system to safely modify a host user interface (UI), as in claim 18 above, but do not disclose wherein the processing unit is operable to request from the extension commands to be inserted at an insert location of a host UI by the insert location's UUID. However, Ezekiel at al. disclose a system that uses universally unique identifier (UUID) or group identifiers to manipulate menu items, like insert locators (i.e. ... group identifiers are used in sorting commands... col 6 lines 50-53). Therefore, it would have been obvious to one having ordinary skill in the art at the time of this invention to use a unique identifier to manipulate menu items, in Padawer et al. One would be motivated to use unique identifiers to manipulate all menu items especially when the application/extension has to work between different menus of host application.

Claim 22: Padawer et al. disclose a computer-accessible medium having components for safely modifying a host user interface with an extension user interface, as in claim 21 above, but do not disclose storing a command with a universal unique identifier (UUID). Furthermore Ezekiel at al. disclose, wherein the command item data structure includes a universal unique identifier (UUID) with which to uniquely identify the command, and the command UI includes at least one of a text, a icon, a status text, and help text, each of which is used to



display the command in the host UI (i.e. ... each entry of command table can contain , for example, name of the command,... and a unique identifier... col 6 lines 34-43). Therefore, it would have been obvious to one having ordinary skill in the art at the time of this invention to store a unique identifier for each command items in CMI array entries, in Padawer et al. One would be motivated to uniquely identify all the command items (menu items) especially when the application/extension has to work between different menus of host application.

Claim 23: Padawer et al. disclose a computer-accessible medium having components for safely modifying a host user interface with an extension user interface, as in claim 21 above, but do not disclose storing insert location with a universal unique identifier (UUID). Furthermore Ezekiel et al. disclose, wherein the insert location data structure includes a universal unique identifier (UUID) with which to uniquely identify the insert location, wherein the insert location represents an actual location within the host UI (i.e. ... each template object corresponds to a window- insert locations- type... col 7 lines 8-16 and Fig. 3B). Therefore, it would have been obvious to one having ordinary skill in the art at the time of this invention to identify window (insert location) with the command items, in Padawer et al. One would be motivated to separate the display part of the menu items from storage part of the command items.

Claim 24: Padawer et al. disclose a computer-accessible medium having components for safely modifying a host user interface with an extension user interface, as in claim 21 above. Furthermore Padawer et al. discloses host to extension (utility customization module) a declarative list having a command item data structure in which to store an extension command UI (Fig. 10A), and an insert location data structure to store an available insert location in a extension UI (Fig. 10B); but do not explicitly disclose the extension interaction with sub-extension and an extension interface to expose available insert locations to sub-extensions that have commands to insert into an extension UI, and to receive a count of command items and command items representing the sub-extension's commands; a sub-extension interface to receive the available insert locations from the extension and to provide the count of command items and command items representing the sub-extension's commands; and an extension process to integrate the sub-extension's command into the extension UI prior to providing to the host the count of command items and command items representing the extension's commands. However, Ezekiel et al. clearly disclose an iterative process of inter-extension communications (i.e. Package to Packet to Template col 8 lines 12-28 and Fig. 4). Therefore, it would have been obvious to one having ordinary skill in the art at the time of this invention to extend an iterative process to inter-extension communications, in Padawer et al. One would be motivated to extend an iterative process to inter-extension communications in

anticipation of moving around or reusing the sub-extensions or sub-menus, as a subset of main menus.

9. Claims 9 and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Padawer et al. (US 5,220,675) and Ezekiel at al. (US 5, 625, 783) in view of Scouten. (US 6,357,038 B1).

Claim 9: Padawer et al. and Ezekiel at al. disclose a method for modifying a host user interface (UI), as in claim 1 above. Furthermore, Padawer et al. disclose,

- further comprising remapping the insert locations into a different portion of the host UI than in prior versions of the host UI (i.e. ... identifier may be automatically generated by the computer... col 6 lines 53-58); and
- integrating the command into the host UI in the remapped insert locations (i.e. ... each entry of command table can contain , for example, name of the command,...and a unique identifier... col 6 lines 34-43);

but Padawer et al. do not disclose a method to keep track of versions. However, Scouten. discloses a system that maps between different versions (i.e. ... a method for producing macros for use by application program ... col 1 lines 24-27). Therefore, it would have been obvious to one having ordinary skill in the art at the time of this invention to keep track of versions of host application as one of the command item attributes, in Padawer et al. One would be motivated to uniquely identify the host application version in all the command items (menu

items) especially when the application/extension have to work between different versions of host application.

Claim 10: Padawer et al. , Ezekiel at al. and Scouten. disclose a method for modifying a host user interface (UI), as in claim 9 above, and, Scouten., further discloses a system that maps between different versions. (i.e. ... a method for producing macros for use by application program ... col 1 lines 24-27).

Therefore, it would have been obvious to one having ordinary skill in the art at the time of this invention to maintain the older version attributes like insert locations, in Padawer et al. One would be motivated to maintain the older version attributes, like insert locations of the host application especially when the application/extensions have to work between different versions of host application.

### ***Response to Arguments***

A. Applicant 's arguments filed on June 14<sup>th</sup>, 2007 have been fully considered but they were found not persuasive.

B. Applicant argues, for Claim 6, "Padawer et al. relies on atoms to store and retrieve strings. Strings and tables of strings are not extensions. They cannot extend the functionality of a host". The examiner respectfully disagrees. Padawer et al. teach

the use of custom menu item (CMI) arrays for manipulating the menu items in the user defined menu area (col 5 lines 3-15). Furthermore, Fig. 10A clearly shows an array that is not just a bunch of strings, but includes command name and attributes, that can extend the host functionality.

C. Applicant argues, for Claim 15, " that an array of command menu items as recited in Padawer et al. is not the same as a memory in which to store a command item representing a UI for a command from an extension ". The examiner respectfully disagrees. Padawer et al. teach the storage, in memory (col 5 lines 59-61), of the CMI array with a structure, including a command (col 5 lines 20-26 and Fig. 10A).

D. Applicant argues, for Claim 21, "that determining a number of menu items as recited in Padawer et al. is not the same as receiving a count of command items and command items representing an extension's commands". The examiner respectfully disagrees. Padawer et al. teach how the user input is received, analyzed to determine the command, and for example, if the command is "ADD", then to create a new entry into CMI array (col 6 lines 41-61 and Fig. 11-12).

E. Applicant argues, for Claim 11, "that the package registration process recited in lines 12-28 of col 8 in Ezekiel et al. and illustrated in Fig. 4, is not the same as an extension returning a number of command items to be inserted into an insert location". The examiner respectfully disagrees. Ezekiel et al. teach that a package

notifies shell 300 of its existence and informs of its command tables and its packets (number of commands), if any, and their respective command tables, if any (col 8 lines 2-11).

F. Applicant argues, for Claim 1, "adding unique identifiers for CMI array entries is not the same as providing unique identifiers for extensions". The examiner respectfully disagrees. Padawer et al. may not explicitly teach adding unique identifiers to the entries 1012 of its CMI array, but it is inherent as the CMI array is "global" (Fig. 10B), meaning that each entry can be uniquely identified. Furthermore, Ezekiel et al. clearly teaches uniquely identifying command items (col 6 lines 33-43), and therefore the examiner maintains that it is obvious to combine these teachings resulting in a uniquely identified extensions (unique CMI array entry).

G. Applicant argues, for Claim 7, "even if Ezekiel et al. was combinable with Padawer et al., which the applicants deny, the resulting combination would not meet the recitations of claims 1 and 7 when combined". The examiner respectfully disagrees, and points out that there is no new grounds for objects, and the previous objections have been answered above.

H. Applicant argues, for Claim 18, "that Ezekiel et al. do not disclose universal unique identifiers, but disclose identifiers that are locally, i.e. within each command table". The examiner respectfully disagrees. Ezekiel et al. teach the structure of an example

application program 260 and show its relationship to add-on software components including add-on component 271 (col 5 lines 37-41). As can be clearly seen in the schematic (Fig. 3A), add-on component 271 is not part of the host application, and hence need some sort of unique identification.

- I. Applicant argues, for Claim 24, "Providing counts is not the same as providing an iterative process". The examiner respectfully disagrees. Ezekiel et al. teach that a package notifies shell 300 of its existence and informs of its command tables and its packets (counts), if any, and their respective command tables, if any (col 8 lines 2-11).
- J. Applicant argues, for Claim 9, "macro file that includes at least two versions of executable configuration code is not the same as remapping the insert locations into a different portion of the host UI than in prior versions of that host UI". The examiner respectfully disagrees. Scouten teaches producing macro files for different operating systems, or could be different versions of same operating system, that permits packaging of the sequence of actions, i.e. insert locations (col 1 lines 24-46 and Figs. 2-5)

### ***Conclusion***

10. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Anil N. Kumar whose telephone number is (571) 270-1693. The examiner can normally be reached on Wednesdays and alternate Mon-Tue and Thu-Fri EST (Alternate Mon-Tue and Thu-Fri off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor Kristine Kincaid can be reached on (571) 272-4063. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.



Art Unit: 2174

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ANK

8/8/2007

*Kristine Kincaid*  
KRISTINE KINCAID  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100